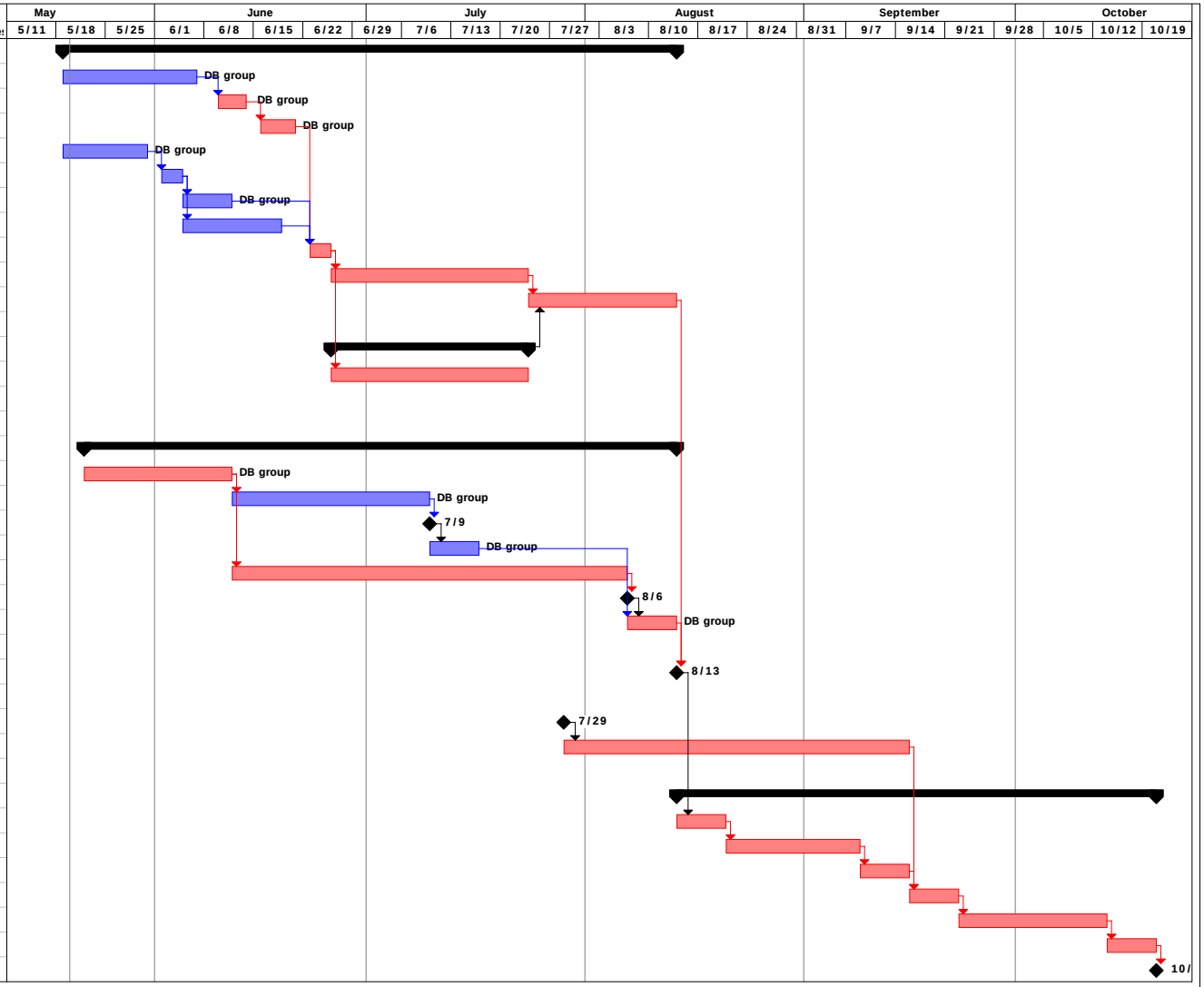
ID	Task Name	Duration	Start	Finish	Predecessors
1	Initial source code conversion	63d	Mon 5/19/97	Wed 8/13/97	
2	ICC: prepare infrastructure packages	3w	Mon 5/19/97	Fri 6/6/97	
3	ICC: infrastructure packages in relea	4d	Tue 6/10/97	Fri 6/13/97	2
4	ICC: release test of examples	5d	Mon 6/16/97	Fri 6/20/97	3
5	ICC: prepare instructions	2w	Mon 5/19/97	Fri 5/30/97	
6	ICC: review instructions	3d	Mon 6/2/97	Wed 6/4/97	5
7	ICC: instruction augmention (option	1w	Thu 6/5/97	Wed 6/11/97	6
8	ICC: Separate existing classes into p	2w	Thu 6/5/97	Wed 6/18/97	6
9	ICC: present to P.Cs.	3d	Mon 6/23/97	Wed 6/25/97	7,8,4
10	ICC: editting & debugging	4w	Thu 6/26/97	Wed 7/23/97	9
11	ICC: special release cycle	3w	Thu 7/24/97	Wed 8/13/97	10,13
12					
13	Define calibration & alignment code	20d	Thu 6/26/97	Wed 7/23/97	
14	Declare initial classes	20d	Thu 6/26/97	Wed 7/23/97	9
15					
16					
17	Database developer infrastructure	60d	Thu 5/22/97	Wed 8/13/97	
18	Database installation prep	3w	Thu 5/22/97	Wed 6/11/97	
19	Database install on ref platforms	4w	Thu 6/12/97	Wed 7/9/97	18
20	Database on reference platforms	0d	Wed 7/9/97	Wed 7/9/97	19
21	Persistant Development Tools releas	5d	Thu 7/10/97	Wed 7/16/97	20
22	Database install at developer sites	8w	Thu 6/12/97	Wed 8/6/97	18
23	Database at developer sites	0d	Wed 8/6/97	Wed 8/6/97	22
24	Test cycle	1w	Thu 8/7/97	Wed 8/13/97	21,23
25					
26	Ready to start persistant conversion	0d	Wed 8/13/97	Wed 8/13/97	24,11
27					
28	Schedule conversion workshop	0d	Tue 7/29/97	Tue 7/29/97	
29	Workshop travel leadtime	7w	Tue 7/29/97	Mon 9/15/97	28
30					
31	Conversion to persistance	48d	Thu 8/14/97	Mon 10/20/97	
32	DDL and makefile conversion to CVS	5d	Thu 8/14/97	Wed 8/20/97	26
33	Edit, debug, problem solve 1	13d	Thu 8/21/97	Mon 9/8/97	32
34	Release, build, test	1w	Tue 9/9/97	Mon 9/15/97	33
35	Workshop	1w	Tue 9/16/97	Mon 9/22/97	34,29
36	Edit, debug, problem solve 2	3w	Tue 9/23/97	Mon 10/13/97	35
37	Release, build, test	1w	Tue 10/14/97	Mon 10/20/97	36
38	First functional release	0d	Mon 10/20/97	Mon 10/20/97	37



Project: Database conversion
Date: Tue 5/20/97

Task | Progress | Summary | Rolled Up Critical Task | Rolled Up Progress
Critical Task | Milestone | Rolled Up Task | Rolled Up Milestone

1 **Initial source code conversion**
This is an "edit level" conversion: int => d_int, insertion of smart pointer classes, wrap of new/delete. At the end of this, we're still dealing with a transient system, but we've done as much preliminary work as possible.

3 **ICC: infrastructure packages in release**
Start date is 6/10/97, a special release. Normal schedule would have been 6/3 or 6/10, neither of which allows time for test & debug before the paris workshop. Duration is set by a normal release build time.

5 **ICC: prepare instructions**
Contains:
- List of needed datatype changes
- Instructions on inserting hints, changes to new and delete
- Instructions on changes to pointers, all types
- Example code in CVS

6 **ICC: review instructions**
Review by selected reconstruction experts for killer problems, not for text presentation, etc

7 **ICC: instruction augmention (optional)**
This task reserves time for adding additional contents to the instructions if they are needed; its optional in the sense that this may not be needed, but elapsed time needs to be reserved.

8 **ICC: Separate existing classes into persistant / transient for first effort**
Reconstruction will go through the entire existing set of classes and decide whether a given one will remain transient or migrate to persistent for the first effort. This may change, of course, but still needs to be done on this timescale.

9 **ICC: present to P.Cs.**
This is the Paris workshop, and the dates are fixed to conform to that.

10 **ICC: editting & debugging**
This has to take place in parallel with normal development on this timescale. In particular, note that this overlaps and uses the same people as the definition of calibration objects, which will need to be persistant on the same time scale

14 **Declare initial classes**
Get an initial set of class declarations in place to form a basis for later editting.

18 **Database installation prep**
Create installation instructions, files and tar files, CD and documentation copies, etc. Distribute to developer sites.

19 **Database install on ref platforms**
Not sure who is really responsible for this

21 **Persistant Development Tools released and tested**
All items needed for basic persistent development, though some are likely to be temporary scaffolds, and appropriate rough documentation.
Includes:
- Database open and close (perhaps via an input module)
- Transaction management; use of and restrictions on "read only"
- How to create and populate a database for debugging
- Tools to "persistant leak checking", rudimentary DB dumps, etc

22 **Database install at developer sites**
Time is difficult to estimate here, but unless we're providing onsite assistance I can't imagine it being much faster than this

24 **Test cycle**
Testing of the infrastructure in released and distributed form, particularly at the remote developer sites.

26 **Ready to start persistant conversion**
This is a decision point

28 **Schedule conversion workshop**
Note that, although no explicit linkage is shown, this needs to be close enough to the end of the edit conversion that some success is guaranteed.

31 **Conversion to persistance**
Although the exact timing of the two common releases during this period are somwhat vague, it clear taht time must be allocated for them.

32 **DDL and makefile conversion to CVS**
Involves freezing (and backing up) CVS, then converting all the .hh files to .ddl files, making the required GNUmakefile changes, and running through a first pass build to create tmp/$BFARCH files. From here, its no longer possible to build the t-only release.

35 **Workshop**
Primary workshop goal is to compare notes on solutions, and identify critical problems.

38 **First functional release**
At this point, the migration has returned us to a functional release equivalent (more or less) to what we had at the beginning of the "Conversion to persistance" composite task. From here, we can start to build on this.